

Logic Programming Engineering – Assignment 5

Dr.-Ing. Sascha Klüppelholz

Exercise 5.1 [Simple arithmetic on lists]

- a) Define a predicate `maxmember/2` that returns the maximum of the elements of a list.

Example:

```
?- maxmember([3,1,5,2], Max).  
Max = 5.
```

- b) Write a predicate `average/2` which computes the average over the numeric elements of a list. For this write an auxiliary predicate `sum/2` for the sum of elements of a list.

Example:

```
?- average([2,3,5,1,6], Av).  
Av = 3.4.
```

Exercise 5.2 [Sieve of Eratosthenes]

Provide a Prolog predicate `eratosthenes/2` that first generates all integers between 2 and a given upper bound and then eliminates all non-primes from the list. Your implementation should follow the idea of Eratosthenes Sieve¹. How many primes are there between 1 and one million? What is the greatest prime lower than one million?

Example:

```
eratosthenes(50,L).  
L = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47].
```

Exercise 5.3 [Subset sum]

Write a predicate `subsetsum/3` that computes for a given list `L` and a given integer number $n \in \mathbb{N}$ a sublist of `L` whose sum over all elements equals to `n`. The predicate should enumerate all solutions.

Examples:

```
?- subsetsum([2,1,3,4], 5, SL).  
SL = [2,3]; SL = [1,4]; false.  
?- subsetsum([-7, -3, -2, 5, 8], 0, SL).  
SL = [-3, -2, 5]; false.
```

¹see, e.g., https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes